APPENDIX A – PROGRAM TO GENERATE HUFFMAN TABLES

```
10   ! RE-STORE "JPEG_PAT"
20     REM *****************************************************
30     REM
40     REM     This is a program to generate Huffman code
50     REM  tables for JPEG compression.  Example tables are
60     REM  generated for common JPEG DC and AC generating
70     REM  values, as well as tables for special byte-aligned
80     REM  JPEG files.
90     REM
100    REM     These correspond to Tables 2, 4, 8, and 9
110    REM  in the patent application specification.
120    REM
130    REM     HP Docket PDNO 100110176
140    REM     November 2001
150    REM
160    REM -----------------------------------------------------
170    REM
180    ALLOCATE Huff_val(255),Huff_bits(15),Huff_size(255)
190    ALLOCATE Huff_code(255),B$[50],Title$[50]
200  ! PRINTER IS "OUTFILE.TXT"
210    PRINTER IS 1
220    REM
230    REM   Generate all four tables
240    REM
250    FOR Table_count=1 TO 4
260       REM
270       REM   Point to correct data
280       REM
290       SELECT Table_count
300       CASE 1
310          RESTORE Table2_data
320          Nval=9
330          Title$="Table 2."
340       CASE 2
350          RESTORE Table4_data
360          Nval=162
370          Title$="Table 4."
380       CASE 3
390          RESTORE Table8_data
400          Nval=9
410          Title$="Table 8."
420       CASE 4
430          RESTORE Table9_data
440          Nval=130
450          Title$="Table 9."
460       END SELECT
470       REM
480       REM   Read table generating values
490       REM
500       FOR I=0 TO Nval-1
510          READ A$
520          Huff_val(I)=DVAL(A$,16)
530       NEXT I
540       READ Huff_bits(*)
550       REM
560       REM   Generate the HUFFSIZE table
570       REM
```

```
580      K=0
590      I=1
600      J=1
610      WHILE (I<=16)
620         WHILE (J<=Huff_bits(I-1))
630            Huff_size(K)=I
640            K=K+1
650            J=J+1
660         END WHILE
670         I=I+1
680         J=1
690      END WHILE
700      Lastk=K
710      Huff_size(K)=0
720      REM
730      REM   Generate the HUFFCODE table
740      REM
750      K=0
760      Size=Huff_size(0)
770      Code=0
780      Flag=0
790      WHILE Flag=0
800         REPEAT
810            Huff_code(K)=Code
820            Code=Code+1
830            K=K+1
840         UNTIL Huff_size(K)<>Size
850         IF Huff_size(K)=0 THEN
860            Flag=1
870            GOTO Skipit! Equivalent of "break" in C
880         END IF
890         REPEAT
900            Code=Code*2
910            Size=Size+1
920         UNTIL Huff_size(K)=Size
930 Skipit: REM
940      END WHILE
950      REM
960      REM   Sort table, format, and print
970      REM
980      CALL Sortem(Lastk,Huff_val(*),Huff_code(*),Huff_size(*))
990      PRINT RPT$(" ",15);Title$
1000     PRINT
1010     FOR K=0 TO Lastk-1
1020        A$=DVAL$(Huff_val(K),16)
1030        A$=A$[7]
1040        B$=DVAL$(Huff_code(K),2)
1050        B$=B$[32-Huff_size(K)+1]
1060        PRINT USING "4D,2X,6A,20A,6D";K,A$,B$,Huff_size(K)
1070     NEXT K
1080     PRINT
1090  NEXT Table_count
1100  STOP
1110  REM
1120  REM   Data sets for generating tables follow
1130  REM
1140 Table2_data: REM
1150  DATA 01,02,00,03,04,05,06,07,08
1160  DATA 00,02,03,01,01,01,01,00,00,00,00,00,00,00,00,00
1170 Table4_data: REM
```

```
1180  DATA 01,02,03,00,04,11,05,12,21,31,41,06,13,51,61,07
1190  DATA 22,71,14,32,81,91,A1,08,23,42,B1,C1,15,52,D1,F0
1200  DATA 24,33,62,72,82,09,0A,16,17,18,19,1A,25,26,27,28
1210  DATA 29,2A,34,35,36,37,38,39,3A,43,44,45,46,47,48,49
1220  DATA 4A,53,54,55,56,57,58,59,5A,63,64,65,66,67,68,69
1230  DATA 6A,73,74,75,76,77,78,79,7A,83,84,85,86,87,88,89
1240  DATA 8A,92,93,94,95,96,97,98,99,9A,A2,A3,A4,A5,A6,A7
1250  DATA A8,A9,AA,B2,B3,B4,B5,B6,B7,B8,B9,BA,C2,C3,C4,C5
1260  DATA C6,C7,C8,C9,CA,D2,D3,D4,D5,D6,D7,D8,D9,DA,E1,E2
1270  DATA E3,E4,E5,E6,E7,E8,E9,EA,F1,F2,F3,F4,F5,F6,F7,F8
1280  DATA F9,FA
1290  DATA 00,02,01,03,03,02,04,03,05,05,04,04,00,00,01,125
1300 Table8_data: REM
1310  DATA 00,01,02,03,04,05,06,07,08
1320  DATA 01,01,01,01,01,01,01,02,00,00,00,00,00,00,00,00
1330 Table9_data: REM
1340  DATA 08,18,28,38,48,58,68,78,88,98,A8,B8,C8,D8,E8,F8
1350  DATA 07,17,27,37,47,57,67,77,87,97,A7,B7,C7,D7,E7,F7
1360  DATA 06,16,26,36,46,56,66,76,86,96,A6,B6,C6,D6,E6,F6
1370  DATA 05,15,25,35,45,55,65,75,85,95,A5,B5,C5,D5,E5,F5
1380  DATA 04,14,24,34,44,54,64,74,84,94,A4,B4,C4,D4,E4,F4
1390  DATA 03,13,23,33,43,53,63,73,83,93,A3,B3,C3,D3,E3,F3
1400  DATA 02,12,22,32,42,52,62,72,82,93,A2,B2,C2,D2,E2,F2
1410  DATA 01,11,21,31,41,51,61,71,81,91,A1,B1,C1,D1,E1,F1
1420  DATA 00,F0
1430  DATA 00,00,00,00,00,00,00,16,16,16,16,16,16,16,16,02
1440  END
1450  SUB Sortem(N,A(*),B(*),C(*))
1460 Sortem: REM
1470     REM ***************************************************
1480     REM
1490     REM     Simple bubblesort routine to sort the arrays
1500     REM  based on A(*).
1510     REM
1520     REM ---------------------------------------------------
1530     REM
1540     FOR I=N-1 TO 1 STEP -1
1550        FOR J=1 TO I
1560           IF A(J)<A(J-1) THEN
1570              T=A(J)
1580              A(J)=A(J-1)
1590              A(J-1)=T
1600              T=B(J)
1610              B(J)=B(J-1)
1620              B(J-1)=T
1630              T=C(J)
1640              C(J)=C(J-1)
1650              C(J-1)=T
1660           END IF
1670        NEXT J
1680     NEXT I
1690  SUBEND
```